# Analisa Perbandingan Kinerja Steganografi pada *Platform Linux* dan *Windows* Menggunakan Kriteria Steganalisis

**Artikel Ilmiah**

**Peneliti :**
**Nugraha Raymond Banoet (672017713)**
**Magdalena A. Ineke Pakereng, M.Kom.**

**Program Studi Teknik Informatika**
**Fakultas Teknologi Informasi**
**Universitas Kristen Satya Wacana**
**Salatiga**
**November 2018**

# Perbandingan Kinerja Steganografi pada *Platform Linux* dan *Windows* Menggunakan Kriteria Steganalisis

**Artikel Ilmiah**

**Diajukan kepada**
**Fakultas Teknologi Informasi**
**untuk memperoleh Gelar Sarjana Komputer**

**Peneliti :**
**Nugraha Raymond Banoet (672017713)**
**Magdalena A. Ineke Pakereng, M.Kom.**

**Program Studi Teknik Informatika**
**Fakultas Teknologi Informasi**
**Universitas Kristen Satya Wacana**
**Salatiga**
**November 2018**

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

# PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : Nugraha Raymond Barcet

NIM : 672017713 Email : nugraha_Raymond@yahoo.com

Fakultas : Teknologi Informasi Program Studi : Teknik Informatika

Judul tugas akhir : Analisa Perbandingan kinerja Steganografi pada Platform Linux dan Windows. Menggunakan Kriteria Steganoesis

Pembimbing : 1. Magdalena A. Ineke Pakereng, M. kom.

2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 18 - Januari - 2019


Tanda tangan & nama terang mahasiswa

Nugraha Raymond Barcet

F-LIB-080

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

# PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : Nugraha Raymond Banoet.
NIM : 672017713 Email : nugraha_raymond@yahoo.com
Fakultas : Teknologi Informasi Program Studi : Teknik Informatika.
Judul tugas akhir : Analisa Perbandingan kinerja steganografi pada platform Linux dan Windows menggunakan kriteria steganalisis.

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

a. ✓ Saya mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA

b. Saya tidak mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* *Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak non-ekslusif kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut.*

** *Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).*

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 18 – Januari – 2019.

Nugraha Raymond Banoet.
*Tanda tangan & nama terang mahasiswa*

Mengetahui,

Magdalena A. Ineke Pakereng, M.kom.
*Tanda tangan & nama terang pembimbing I*

*Tanda tangan & nama terang pembimbing II*

F-LIB-081

## Lembaran Pengesahan

| | |
|---|---|
| Judul Tugas Akhir | : Analisa Perbandingan Kinerja Steganografi pada *Platform Linux* dan *Windows* Menggunakan Kriteria Steganalisis |
| Nama Mahasiswa | : Nugraha Raymond Banoet |
| NIM | : 672017713 |
| Program Studi | : Teknik Informatika |
| Fakultas | : Teknologi Informasi |

Disetujui oleh,

Magdalena A. Ineke Pakereng, M.Kom.
Pembimbing

Diketahui oleh,

Wiwin Sulistyo, ST., M.Kom.
Dekan

Magdalena A. Ineke Pakereng, M. Kom.
Ketua Program Studi

Dinyatakan Lulus Tanggal : 28 November 2018

*Reviewer* :

Prof. Ir. Danny Manongga, M.Sc., Ph.D.

**Pernyataan**

Artikel Ilmiah berikut ini :

Judul : Analisa Perbandingan Kinerja Steganografi pada *Platform Linux* dan *Windows* Menggunakan Kriteria Steganalisis

Pembimbing : Magdalena A. Ineke Pakereng, M.Kom.

adalah benar hasil karya saya :

Nama : Nugraha Raymond Banoet

NIM : 672017713

Saya menyatakan tidak mengambil sebagian atau seluruhnya dari hasil karya orang lain kecuali sebagaimana yang tertulis pada daftar pustaka.

Pernyataan ini dibuat dengan sebenar-benarnya sesuai dengan ketentuan yang berlaku dalam penulisan karya ilmiah.

Salatiga, 21 November 2018

Nugraha Raymond Banoet

## 1. Pendahuluan

Teknologi merupakan sarana pendukung untuk memenuhi kebutuhan hidup manusia, karena dengan adanya teknologi manusia dapat menghemat waktu untuk bertukar informasi berupa data teks, gambar, *audio* dan *video* menggunakan teknologi internet yang memudahkan pengguna untuk bertukar informasi dengan waktu yang singkat ditambah lagi teknologi informasi yang terus berkembang setiap saat. Berbagai serangan berupa *spam, virus* penyadapan maupun *hacker* semakin berkembang, internet tidak menjamin akan keamanan informasi. Berbagai keamanan data terus dikembangkan diantaranya kriptografi dan steganografi.

Teknik pengamanan pesan rahasia menggunakan steganografi merupakan bagian dari kriptografi. Kriptografi adalah teknik merubah informasi asli menjadi informasi acak yang tidak dapat dimengerti pihak lain, namun informasi yang terlihat acak akan menimbulkan kecurigaan untuk itu pesan akan dikirimkan menggunakan steganografi dimana informasi akan disisipkan kedalam citra digital dengan begitu pihak lain akan sulit untuk mengenali keberadaan pesan rahasia [1].

Steganografi dan steganalisis. Algoritma steganografi terus dikembangkan untuk menyembunyikan keberadaan pesan rahasia diantaranya adalah *A Graph Theory Approach*, sementara algoritma steganalisis juga terus berkembang untuk mendeteksi keberadaan pesan rahasia yang disembunyikan dengan algoritma-algoritma steganografi yang telah ditemukan [2]. Steganografi akan menggunkan aplikasi *steghide* dan kemudian akan dianalisis menggunakan metode yang ada dengan kriteria-kriteria yang ditentukan.

Penelitian ini membahas bagaimana menganalisis kinerja data citra steganografi kemudian hasilnya akan di uji meggunkan kriteria steganalisis, yaitu, *Fidelity*: melihat mutu citra secara indrawi. *Robustness*: Menguji ketahanan citra dengan serangan seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, *cropping*, enkripsi, dan sebagainya. *Recovery*: data hasil steganografi harus dapat diambil kembali dalam keadaan seperti semula sebelum disembunyikan. [3] Pengujian akan dilakukan pada data citra gambar beruba *file* jpg dan data citra *audio* berupa *file* wav dengan menggunakan aplikasi yang ada maupun aplikasi yang akan dibuat pada *platform linux* dan *windows* sehingga memudahkan pengguna untuk dapat mengetahui kinerja steganografi serta menganalisisnya pada kedua sistem operasi tersebut.

## 2. Tinjauan Pustaka

Peneletian terkait sebelumnya pernah dilakukan dengan judul " Steganalisis Citra Digital menggunakan Metode *Discrete Wavelet Transform* dan K-*Nearest Neighbor*. Penelitian tersebut membahas tentang bagaimana menganalisis data citra strganografi menggunakan *preprocessing*: proses untuk tranformasi citra RGB ke citra *Gray*, ekstrasi ciri: dengan menghitung *mean*, *standar deviation*, *kurtosis*, dan *skewness*, klasifikasi: dengan menggunakan algoritma KNN (*K-Nearest Neighboor*), dan verifikasi: untuk menguji akurasi

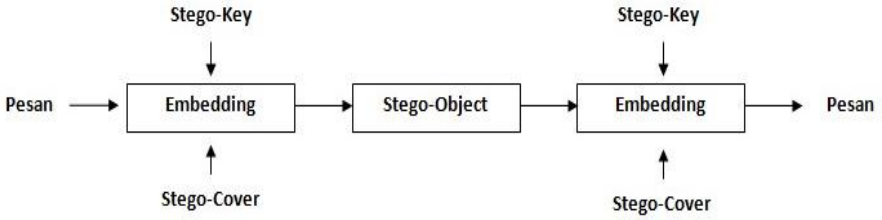
citra dengan sistem steganalisis yang ada. Hasil penelitian tersebut adalah sebuah aplikasi steganalisis yang dapat membedakan citra steganografi dan *non* steganografi [4].

Pada penelitian yang berjudul "Analisis Steganografi Citra Digital menggunakan Metode *Spread Spectrum* berbasis Android". Hasil dari penelitian tersebut adalah menganalisis *embedded* dan *embedding* citra steganografi dengan melihat perbedaan MSE dan PSNR, *noise*, *crooping*, *robustness* dengan metode *spread spectrum* menghasilkan performansi *imperceptibility* antara citra *cover* dan citra *stego* sangatlah mirip. Ukuran citra rahasia yang disisipkan pada citra *cover* mempengaruhi waktu penyisipan dan ekstraksi. Semakin besar ukuran citra rahasia yang disisipkan maka semakin lama waktu yang diperlukan untuk penyisipan dan ekstraksi. Ukuran citra rahasia yang disisipkan mempengaruhi kualitas citra *stego*. Semakin besar ukuran citra rahasia yang disisipkan maka semakin besar pula nilai MSE dan BER yang didapat sehingga kualitas citra *stego* semakin turun. Citra *stego* tidak tahan terhadap serangan *noisegaussian*, serangan ini pada citra stego di pengaruhi oleh nilai standar deviasi (*sigma*). Semakin besar nilai *sigma* maka semakin besar pula nilai MSE dan BER sehingga tingkat kesalahan/*error* semakin besar. Citra *stego* tidak tahan terhadap serangan *cropping* dengan meng-*crop* ukuran *pixel* citra sebesar 0,25 atau 0,5 kali ukuran *pixel* citra *stego*, kemudian dikembalikan lagi dengan cara melakukan proses *resize* ke ukuran semula. Semakin besar ukuran *rasio cropping* maka semakin besar nilai MSE dan BER sehingga semakin kecil pula nilai PSNR. Citra *stego* tidak tahan terhadap serangan *compress* dengan cara meng-*compress* citra berdasarkan kualitas kompresinya yaitu sebesar 10%, 50%, dan 100%. Semakin besar kualitas kompresinya maka semakin kecil nilai MSE dan BER sehingga semakin besar pula nilai PSNR [4].

Berdasarkan penelitian-penelitian yang sudah dilakukan membahas tentang analisis perbandingan citra sebelum dan sesudah steganografi dengan hasil sebuah aplikasi steganalisis, pada penelitian ini akan dilakukan analisis perbandingan kinerja steganografi yang terdapat pada *platform windows* dan *linux* dengan membandingkan data citra dari masing-masing *platform*.

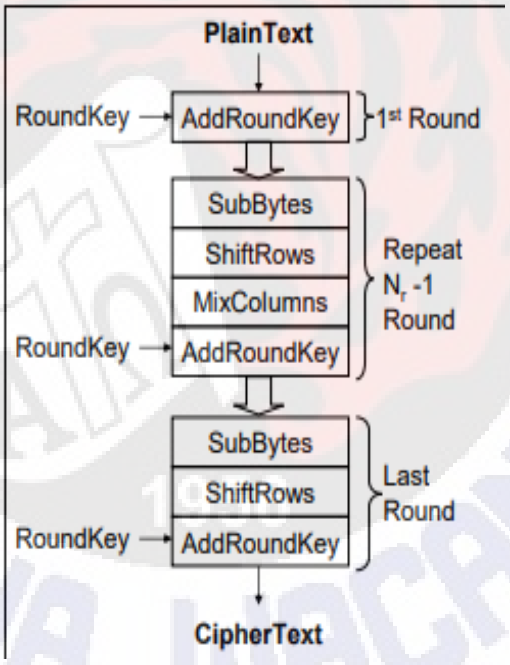
*Steghide* merupakan salah satu program yang mengimplementasikan teknik penyembunyian pesan yang tahan terhadap *Stegdetect*. Program ini dirancang oleh Stefan Hetzl untuk menyembunyikan beragam *file* dalam format gambar dan *audio*. Program ini mengimplementasikan *stegosystem* yang tahan terhadap serangan *first order statistical test*. *Steghide* dibangun sebagai *open source* oleh karena itu dapat secara bebas dimodifikasi dan didistribusikan di bawah GNU *General Public Licence*. Dengan demikian *steghide* berpotensi untuk terus dikembangkan oleh siapa saja. Saat ini *Steghide* telah mencapai versi 0.5.1 dan dapat dijalankan pada sistem operasi *unix* dan *windows* [3].

Pesan steganografi terlihat dengan rupa lain seperti gambar, artikel, atau pesan-pesan lainnya. pesan yang tertulis ini merupakan tulisan yang menyelubungi atau menutupi. tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi. [5]



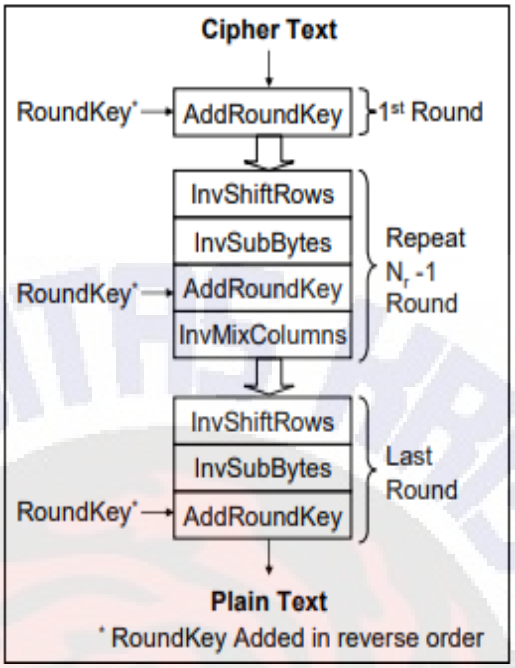

**Gambar 1** Proses Kerja Steganografi [6]

Algoritma *Rijndael* merupakan algoritma yang bekerja pada aplikasi *steghide* ditetapkan oleh NIST sebagai AES pada bulan Oktober 2000. Algoritma *rijndael* ditemukan oleh Vincent Rijmen dan Joan Daemen dari Belgia. *Rijndael* termasuk dalam algoritma kriptografi yang sifatnya simetris dan *block cipher*. *Rijndael* mendukung panjang kunci 128 bit, 192 bit, dan 256 bit. Panjang kunci dan ukuran *block* dapat dipilih secara independen. Pada Gambar 2, ditunjukan proses enkripsi *rijndael* [7]




**Gambar 2** Proses Enkripsi *Rijndael* [7]

Proses enkripsi pada algoritma *rijndael* terdiri dari 4 jenis transformasi *byte*, yaitu *SubBytes(), ShiftRows(), MixColumns(),* dan *AddRoundKey()*. Pada awal proses enkripsi, masukan yang telah berbentuk *array state* akan mengalami transformasi *AddRoundKey()*. Setelah itu, *array state* akan mengalami transformasi *SubBytes(), ShiftRows(), MixColumns(),* dan *AddRoundKey()* secara berulang-ulang sebanyak *Nr*. Proses ini dalam algoritma *rijndael* disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya di mana pada *round* terakhir, *array state* tidak mengalami transformasi *MixColumns()* [7].




**Gambar 3** Proses Dekripsi *Rijndael* [7]

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher*. Transformasi yang digunakan pada *inverse cipher* adalah *InvShiftRows(), InvSubBytes(), InvMixColumns()*, dan *AddRoundKey()* [7].

*A graph theory approach* merupakan algoritma yang dikembangkan oleh Stefan Hetzl bersama Peter Mutzel. Latar belakang dari pembentukan algoritma baru ini adalah bahwa *stego-system* yang ada saat itu telah tidak aman karena pesan rahasia yang ada pada *stego-data*, berformat jpeg khususnya, telah dapat dikenali keberadaannya. Teknik steganalisis yang digunakan salah satunya menggunakan *firstorder statistic*. Perubahan *first-order statistic* menandakan bahwa terjadi perubahan dari gambar asli terhadap *stego-data*. Sehingga keberadaan pesan rahasia dalam *file* gambar berformat jepg dapat diketahui. Melalui proses analisis selanjutnya misalnya melaui *brute-force dictionary attack* pesan rahasia pun dapat diketahui. Oleh karena itu, menurut pemikiran Stefan Hetzl dan Peter Mutzel, diperlukan algoritma baru yang tahan terhadap serangan dengan *first-order statistic test*. Paradigma *stego-system* yang ada saat itu adalah pesan rahasia disusun sedemikian rupa sehingga menimpa *pixel* pada *cover-data*. Cara ini mengakibatkan perubahan cukup besar pada *first-order statistic* gambar.

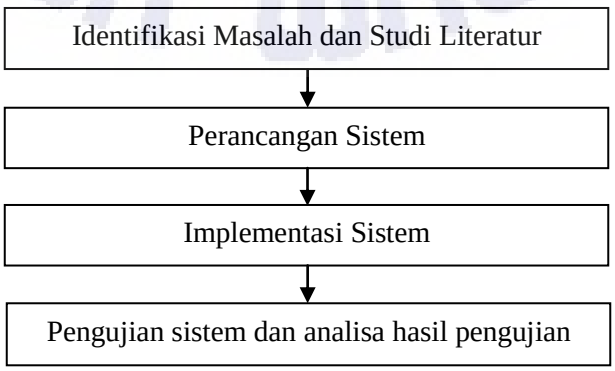
*A Graph Theory Approach* kemudian dirancang dengan paradigma yang berbeda. *Pixel-pixel* pada gambar diganti dengan komponen data pesan rahasia. Akibatnya tidak terjadi perubahan *first-order statistic* pada gambar. Secara garis besar, proses yang terjadi adalah [3] :

- *Pixel-pixel* yang akan dimodifikasi direpresentasikan sebagai puncak atau *verteks*.
- Setiap *pixel* akan dipasangkan dengan *pixel* yang sesuai. Setiap pasangan yang mungkin dihubungkan oleh garis/sisi dengan *pixel* tersebut.
- Pesan rahasia disembunyikan ke dalam pixel-*pixel* tersebut dengan cara penyelesaian komputasi *kombinatorial* dengan menghitung pencocokan kardinalitas maksimum.
- Hasil perhitungan digunakan sebagai penentu dimanakah pesan rahasia akan menggantikan pasangan *pixel* yang sesuai.
- Untuk meminimalisasi perubahan secara visual, setiap sisi dikenai bobot tertentu. Selain tahan terhadap *first-order statistical test*, kelebihan dari algoritma ini adalah tidak adanya kebergantungan sistem pada tipe *cover-data* yang digunakan (misalnya gambar, *audio,..*) [3].

Kriteria yang digunakan untuk pengujian diantaranya. 1) *Fidelity*. Mutu *cover-object* tidak banyak berubah setelah disisipi *embedded message*. Secara indrawi pengamat dapat membedakan *cover-object* dan *stego-object*; 2) *Robustness*. Data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung (seperti pengubahan kontras, penajaman, pemampatan, *rotasi*, perbesaran gambar, pemotongan (*cropping*), enkripsi, dan sebagainya). Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak; ) *Recovery*. Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*). Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut [4].

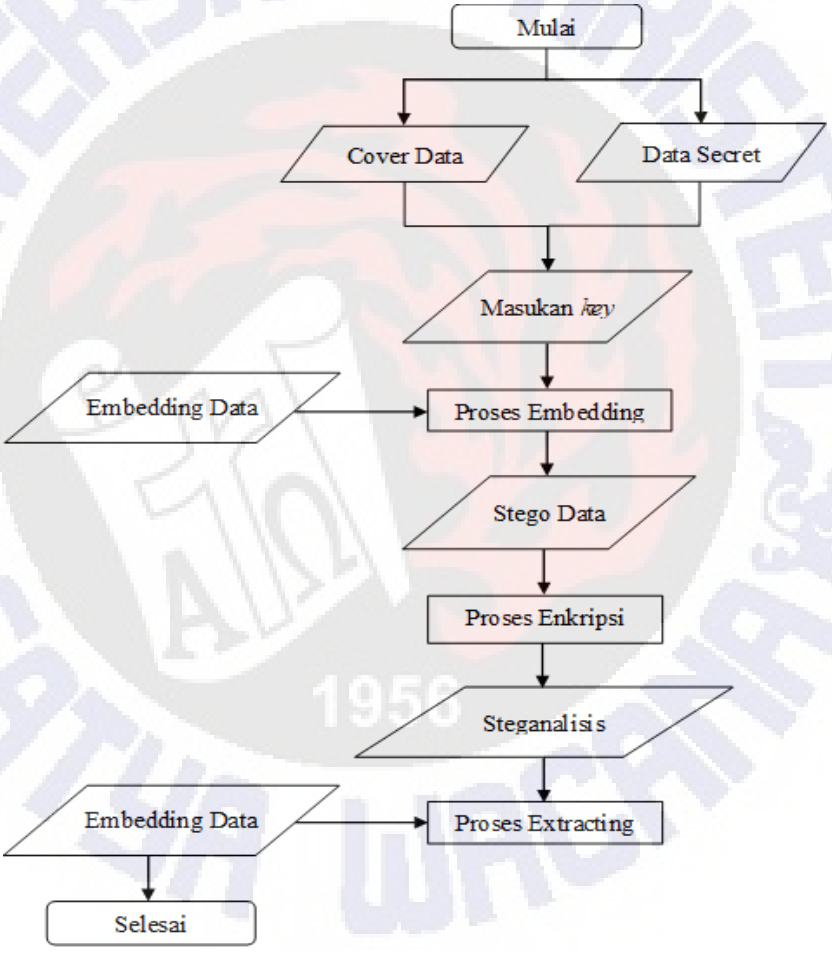
## 3. Metode Penelitian dan Perancangan Sistem

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam empat tahapan, yaitu: (1) Analisis kebutuhan dan pengumpulan data, (2) Perancangan sistem, (3) Implementasi sistem yaitu Perancangan aplikasi/program, dan (4) Pengujian sistem serta analisis hasil pengujian.



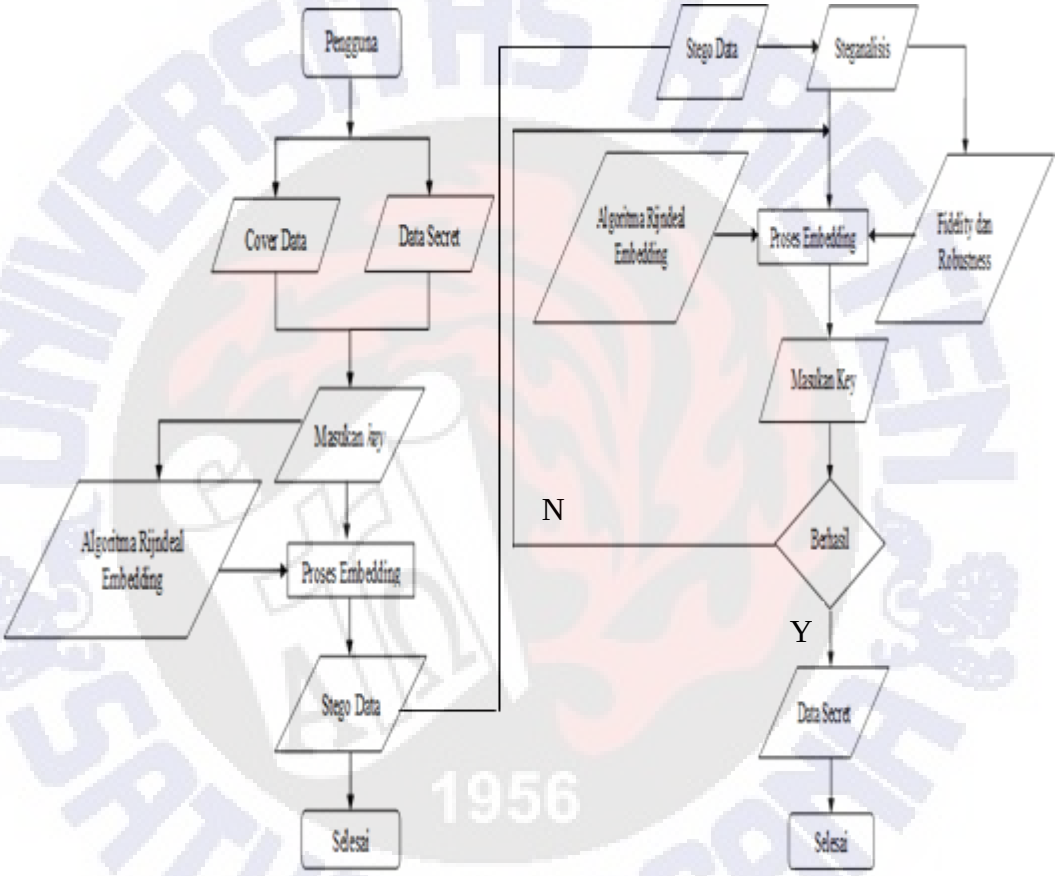

**Gambar 4** Tahapan Penelitian

Tahapan penelitian pada Gambar 4, dapat dijelaskan sebagai berikut. *Tahap pertama*: identifikasi masalah, dilakukan analisis terhadap permasalahan yang ada, yaitu keamanan data dengan proses analisis *citra-data* dan *stego-data*; *Tahap kedua*: perancangan sistem, setelah dilakukan analisa terhadap permasalahan yang ada, maka langkah selanjutnya adalah uji ketahanan *stego-data* terhadap serangan; *Tahap ketiga*: implementasi sistem, yaitu menguji dengan aplikasi yang ada atau membuat aplikasi sesuai perancangan proses pada *tahap kedua* dan *Tahap keempat*: pengujian sistem dan analisa hasil pengujian, yaitu dilakukan proses pengujian terhadap hasil rancanga dan melihat solusi terhadap masalah yang teridentifikasi. Alur proses steganografi akan ditunjukan pada Gambar 5.




**Gambar 5** Alur Sistem Steganografi

Pada Gambar 5 telihat alur kerja steganografi, pada tahap data *cover* terdapat dua jenis *file* yang akan dilakukan penyisipan data *secret* yaitu, *file* jpg dan wav dan masing-masing *file* akan menghasilkan *stego-data* yang terdapat data *secret* di dalamnya. Selanjutnya pada tahap steganalisis *file* akan di uji dengan tiga kriteria steganografi yaitu, *fidelity, robustness* dan *recovery* dengan proses *extracting* untuk membuktikan apakah *stego-data* tahan terhadap manipulasi dengan melihat hasil *extraction* pada *platform windows* dan *linux*. Proses *embedding* akan ditujukan pada Gambar 6 dan proses *extracting* pada Gambar 7.




**Gambar 6** Proses *Embedding*

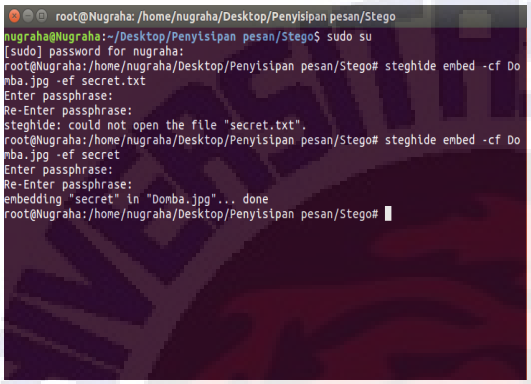



**Gambar 7** Proses *Extracting*

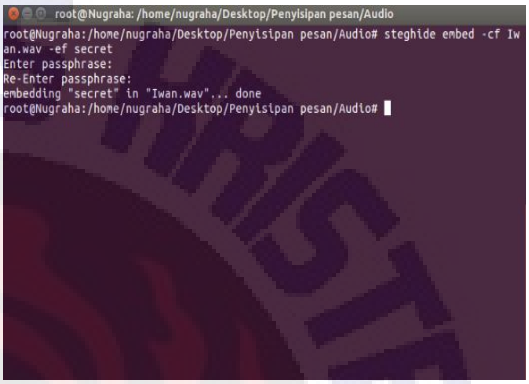
Pada Gambar 6 menjelaskan proses *embedding*, pengguna meng*input cover data* dan *data secret* untuk digabungkan, *cover data* sebagai wadah untuk *data secret* kemudian akan di *embedding* dengan *key* menggunakan algoritma *rijndael* dan akan menghasilkan *Stego Data*. Selanjutnya Gambar 7 menunjukan proses *extracting*, *Stego Data* akan di uji dengan steganalisis menggunakan *fidelity* dan *robustness* dan proses *embedding* untuk menemukan *data secret* jika tidak berhasil maka akan dilanjutkan proses *extracting* tanpa steganalisis

## 4. Hasil dan Pembahasan

Pada bagian ini dijelaskan hasil penelitian yang dilakukan. Aplikasi yang digunakan dalam penelitian ini adalah *steghide* yang terdapat pada *platform linux* dan dijalankan menggunakan *command* di *terminal* untuk *embedding* data steganografi. berikut dilakukan penyisipan dengan dua jenis citra, citra format jgp dan citra wav.




**Gambar 8** Proses *Embedding* Data jpg

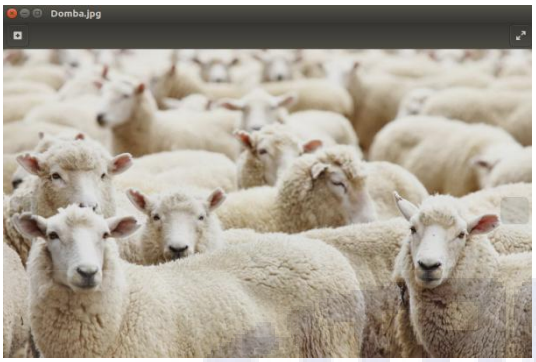



**Gambar 9** Proses *Embedding* Data wav

Pada Gambar 8 proses *embedding* dengan citra gambar Domba.jpg yang akan disisipkan pesan rahasia *secret*, kemudian diminta masukan *enter passphrase* untuk keamanan data citra, setelah itu citra berhasil disembunyikan. Pada Gambar 9 proses *embedding* dengan citra *audio* Iwan.wav sebagai tempat penyisipan pesan *secret*, setelah diminta masukan *enter passphrase* untuk keamanan data citra, selanjutnya citra berhasil disembunyikan.

Pengujian terhadap *fidelity* dapat dianggap sebagai pengujian apakah *steghide* tahan terhadap *detection attack*. *Detection attack* merupaka serangan yang bertujuan untuk mengetahui keberadaan pesan rahasia pada suatu *file*. *Detection attack* terdiri dari RQP *attack*, RS *attack*, serta *know cover attack*. Pengujian terhadap *fidelity* ini lebih khusus pada *known-cover attack* yang artinya pihak penyerang diasumsikan mempunya gambar asli sesuai *cover*-data yang digunakan. Pengujian dilakukan dalam dua tahap pada *format file* yaitu jpg dan wav.

Pengujian pertama: Membandingkan tampilan visual pada jpg yaitu, *cover-data* dan *stego-data.* Terlihat perbandingan tampilan gambar, ditunjukan pada Gambar 10 dan Gambar 11.




**Gambar 10** *Cover-data* Domba.jpg

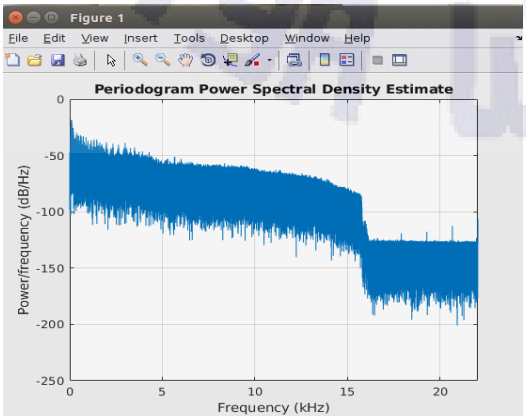

**Gambar 11** *Stego-data* Domba.jpg

Secara kasa mata, kedua Gambar tersebut tidak ada perbedaan antara kedua Gambar *citra-data* dan *stego-data*.

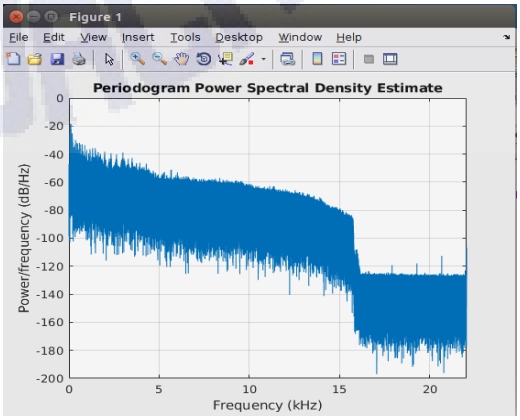

**Gambar 12** *Cover-data* Domba.jpg


**Gambar 13** *Stego-data* Domba.jpg

Melalui Gambar 12 dan Gambar 13 terlihat jelas bahwa terdapat perbedaan dan terjadi pergeseran antara *cover-data* dan *stego-data* pada saat di perbesar dengan *magnify* 2x. Pengujian selanjutnya pada *file* wav ditunjukan pada Gambar 14 dan Gambar 15.




**Gambar 14.** *Cover-data* Iwan.wav




**Gambar 15.** *Stego-data* Iwan.wav

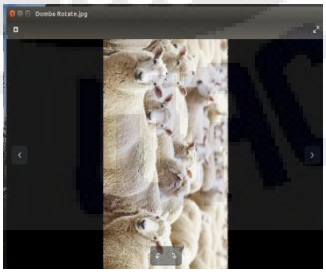
Pengujian pertama juga telah dilakukan pada *cover-audio* yang telah dilakukan terhadap *file* wav. *Cover-data* masih berupa *file* asli *audio* sedangkan *stego-data* berupa *file* steganografi *audio*. Hasilnya, tetap terdapat perbedaan secara visual antara *stego-data* dan *cover-data*. Sedangkan pada percobaan antar *file* wav, yaitu Iwan.wav sebagai *cover-data* yang disispikan data rahasia. Bahwa tidak terdapat perubahan yang dapat ditangkap telinga manusia antara *cover-data* dan *stego-data.* Keluaran suara *cover-data* sama persis dengan *stego-data*. Hasil percobaan berupa histogram sinyal suara pada *file* wav yang di uji terdapat pada Gambar 14 dan Gambar 15. Pada gambar sinyal suara sulit untuk mengenali perbedaan antara *cover-data* dan *stego-data* tetapi kalau dilihat dengan teliti terdapat perubahan *range* pada *power*/*frequency (dB/Hz), cover-data* dengan *range* -250 dan *stego-data* dengan *range* -250 karena ada terjadi perubahan *frequency.*

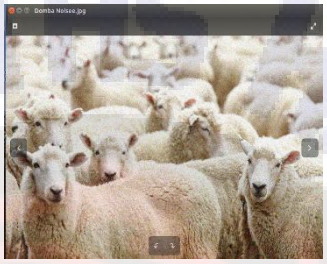
Pengujian ke dua: Membandingkan ukuran *file cover-data* dan *stego-data.* Setelah menguji *fidelity* pada tampilan visual dan *audio*. C*over-data* Domba.jpg memiliki ukuran *file* sebesar 117.054*bytes*, sedangkan *stego-data* 119.482*bytes*. Terdapat penambahan *bytes* berjumlah 2.*428bytes* setelah proses penyisipan. Jika pihak lain memiliki *file* asli maka akan terdapat kecurigaan.

Perbandingan ukuran pada *file audio, cover-data* Iwan.wav memiliki ukuran *file* yang tetap yaitu 52.185.767*bytes*, sebelum dan sesudah penyisipan ukuran kedua *file* tidak ada perbedaan, dengan demikian penyembunyian pesan dengan *file* wav tidak bisa diketahui melalui ukuran *file*.

Pengujian terhadap *robustness* dilakukan bersamaan dengan *recovery*, karena kedua hal tersebut saling berkaitan. *File* dengan format yang berbeda akan dilakukan dengan manipulasi yang berbeda pula, akan dilakukan pengujian pada *file* jpg dan wav dengan tiga jenis manipulasi, ditunjukan pada Tabe l.

**Tabel 1** Pengujian Manipulasi *File* jgp

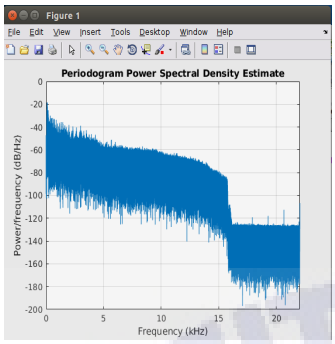
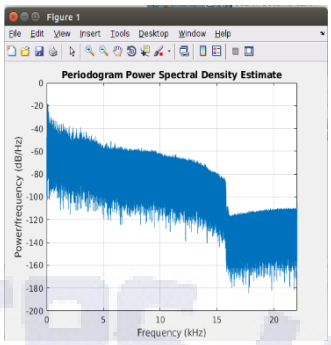
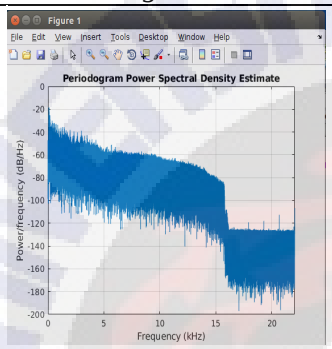
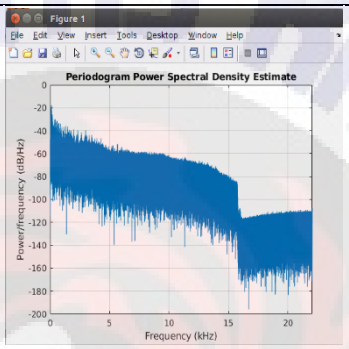
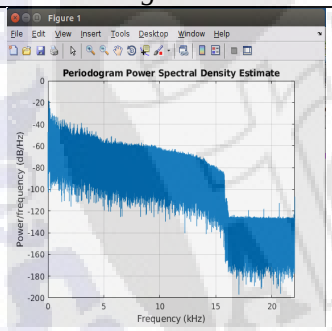
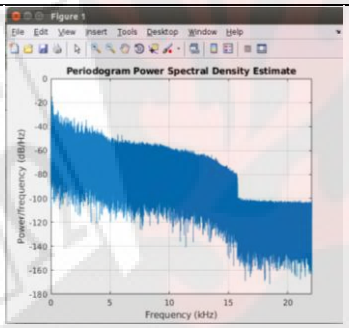
| No | Manipulasi Gambar *Robustness* | | *Recovery* | Hasil |
|---|---|---|---|---|
| | Sebelum | Sesudah | | |
| **1.** |  *Stego-data* |  *rotate* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |

| | | | | |
|---|---|---|---|---|
| **2.** |  *stego-data* |  *Cropping* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |
| **3.** |  *Stego-data* |  *Noise* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |

Pengujian ketiga: Sebelum dilakukan manipulasi dari *file* Domba.jpg akan menghasilkan pesan teks didalamnya yaitu, *secret*.txt. Setelah dilakukan manipulasi yang terlihat pada Tabel 1 bagian pertama dengan pengujian *rotate* terlihat pada menu *recovery file* tidak berhasil di *extract* dengan perintah, *"extracting data…steghide : could not extract any data with that passphrase*" aritnya pesan tidak dapat di *extract*, maka *file* hasil manipulasi dinyatakan tidak berhasil. Bagian kedua dengan pengujian *cropping* terlihat pada menu *recovery file* tidak berhasil di *extract* dengan perintah, *"extracting data…steghide : could not extract any data with that passphrase*" aritnya pesan tidak dapat di *extract*, maka *file* hasil manipulasi dinyatakan tidak berhasil. Bagian ketiga dengan pengujian *noise* terlihat pada menu *recovery file* tidak berhasil di *extract* dengan perintah, *"extracting data…steghide : could not extract any data with that passphrase*" Hasil dari Tabel 1 pada bagian 1 sampai bagian 3 tidak terdapat perbedaan perintah, maka *file stego-data* tidak berhasil di *extract* setelah dilakukan manipulasi. Proses pengujian terhadap *file* wav ditunjukan pada Tabel 2.

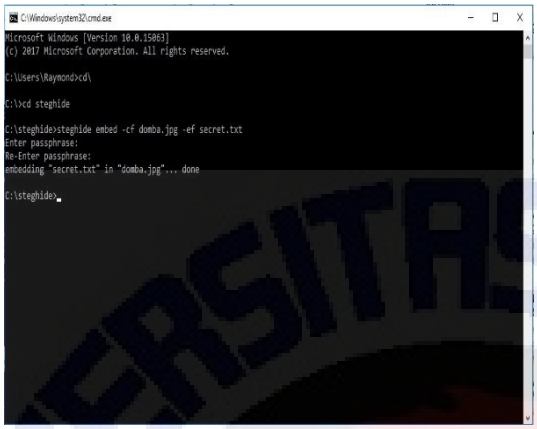
**Tabel 2** Pengujian Manipulasi *File* wav

| **No** | **Manipulasi Gambar *Robustness*** | | ***Recovery*** | **Hasil** |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | | |

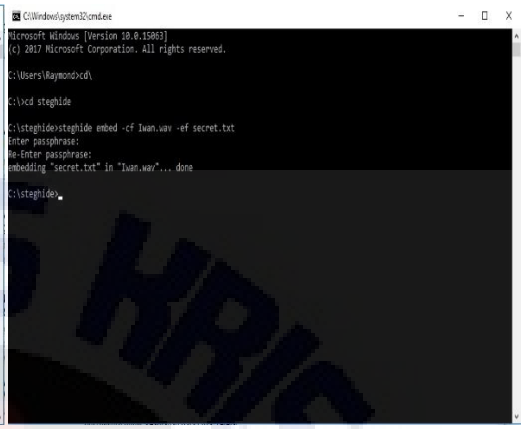
| | | | | |
|---|---|---|---|---|
| 1. |  *Stego-data* |  *Reverse* | *Steghide: could not extract any data with that passphrase* | Tidak berhasil |
| 2. |  *stego-data* |  *Silent* | *Steghide: could not extract any data with that passphrase* | Tidak berhasil |
| 3. |  *Stego-data* |  *Echo* | *Steghide: could not extract any data with that passphrase* | Tidak berhasil |

Pengujian ketiga juga dilakukan pada *file* wav sebelum adanya serangan *file* pada media wav terdapat pesan rahasia yaitu, *secret.txt*. Pada Tabel 2 bagian yang pertama terlihat adanya serangan *reverse* pada menu *recover* adalah perintah yang dikeluarkan *steghide* ketika *stego-data* di-*reverse* bahwa citra tidak dapat di *extract* dan selanjutnya citra tidak berhasil di *extract*. Pada bagian kedua serangan berupa *silent* pada *stego-data* sama dengan bagian yang pertama bahwa citra tidak dapat di *extract*, hal yang sama terjadi pada bagian ketiga dengan serangan berupa *echo* dan data tidak dapat di *extract*.

Selanjutnya *steghide* akan dijalankan pada *platform linux* dan akan dilakukan pengujian yang sama seperti pada *windows*.



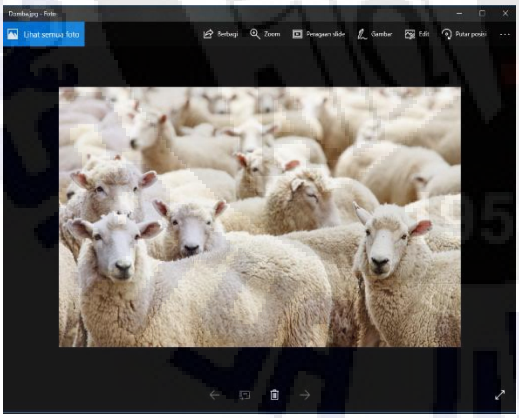

**Gambar 16** Proses *Embedding File* jpg



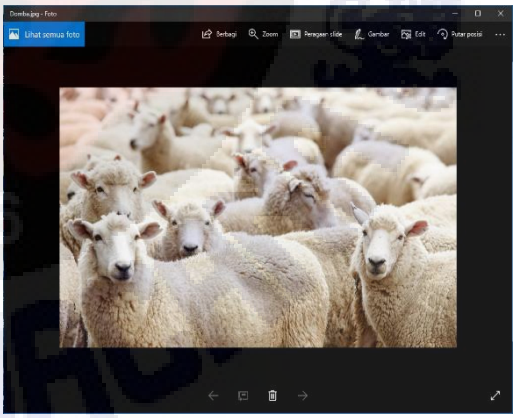

**Gambar 17** Proses *Embedding File* wav

Pada Gambar 16 dan Gambar 17 ditunjukan proses *embedding* dengan citra gambar Domba.jpg dan citra *audio* Iwan.wav yang akan disisipkan pesan rahasia *secret* pada masing-masing citra, setelah itu akan diminta masukan *enter passphrase* untuk keamanan *cover-data* dan citra berhasil disembunyikan.

Pengujian pertama: Membandingkan tampilan visual pada jpg yaitu, *cover-data* dan *stego-data*. Proses perbandingan terlihat pada Gambar 18 dan Gambar19.




**Gambar 18** *Cover-data* Domba.jpg



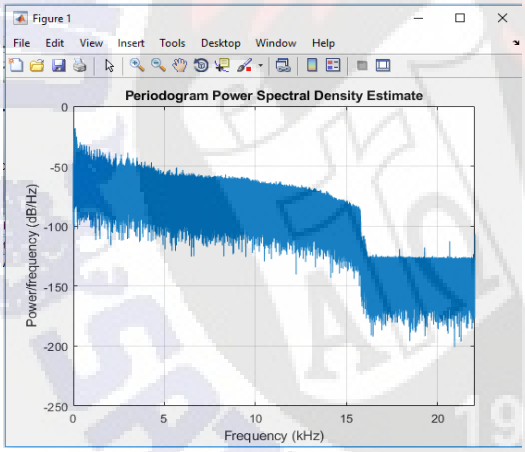

**Gambar 19** *Stego-data* Domba.jpg

Pada Gambar 18 terlihat tampilan gambar *cover-data* sebelum dilakukan penyisipan dan Gambar 19 *stego-data* setelah dilakukan penyisipan. Tidak ditemukan adanya perbedaan sebelum dan sesudah penyisipan.

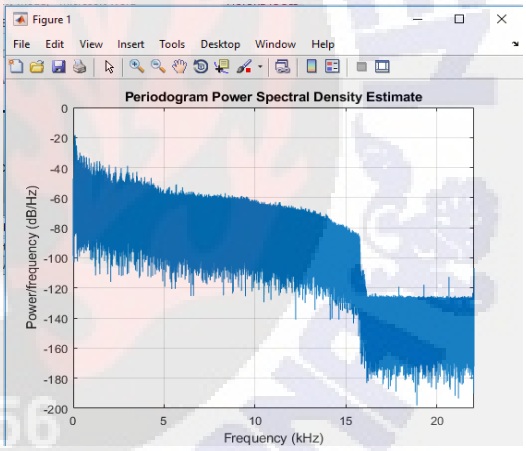
**Gambar 20** *Cover-data* Domba.jpg

**Gambar 21** *Stego-data* Domba.jpg

Membandingkan tampilan visual pada jpg yaitu, *cover-data* dan *stego-data*. Terlihat pada kedua gambar tidak terdapat perbedaan. Melalui Gambar 20 dan 21, secara kasat mata terlihat bahwa, tidak ditemukan adanya perbedaaan. Kedua gambar tersebut terlihat sama ketika di perbesar 2X, berbeda ketika menggunakan *linux* Perbandingan histogram citra data wav ditunjukan pada Gambar 22 dan Gambar 23.




**Gambar 22** *Cover-data* Iwan.wav




**Gambar 23** *Stego-data* Iwan.wav

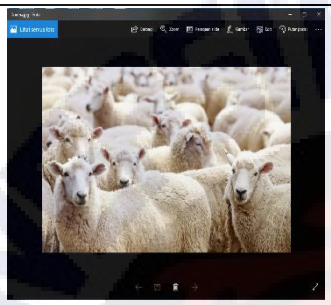
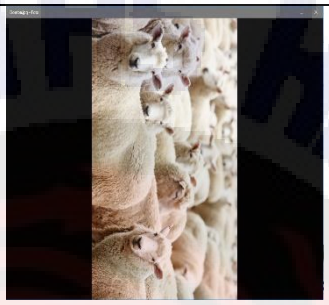
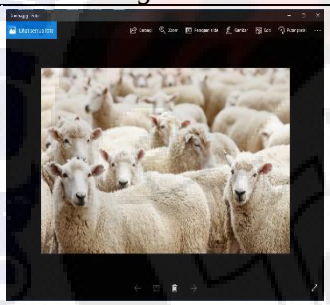
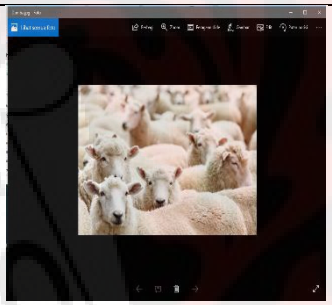
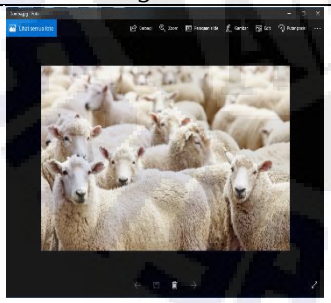
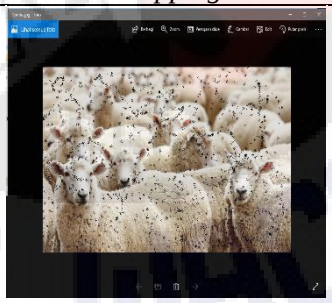
Pengujian pertama juga dilakukan pada *file* wav dengan mendengarkan suara pada kedua citra data, bahwa tidak terdapat perbedaan suara pada kedua *file* tersebut, jika dilihat dari histogram kedua *file* wav, maka dari kedua Gambar 22 dan Gambar 23 ditemukan perbedaan *range cover-data* dengan *range* -250 dan *stego-data* dengan *range* -200.

Pengujian ke dua: Membandingkan ukuran *file cover-data* dan *stego-data*. Setelah menguji *fidelity* pada tampilan visual dan *audio*. C*over-data* Domba.jpg memiliki ukuran *file* sebesar 117.054*bytes*, sedangkan *stego-data* 136.479*bytes*. Terdapat penambahan *bytes* berjumlah 19.425*bytes* setelah proses penyisipan. Jika pihak lain memiliki *file* asli maka akan terdapat kecurigaan.

Perbandingan ukuran pada *file audio*, *cover-data* Iwan.wav memiliki ukuran *file* yang tetap yaitu 52.185.767*bytes*, sebelum dan sesudah penyisipan

kedua *file* tidak ada perbedaan, dengan demikian penyembunyian pesan dengan *file wav* tidak bisa diketahui melalui ukuran *file*, tetapi terlihat perbedaan pada histogram. Selanjutnya tampilan perbandigan *file* gambar ditunjukan pada Tabel 3.

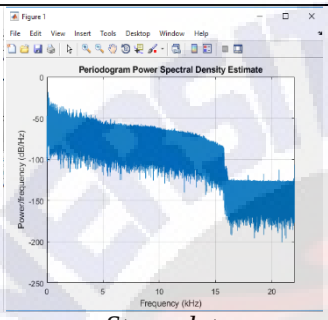
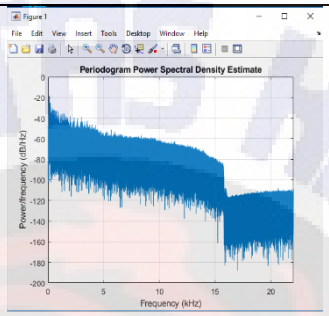
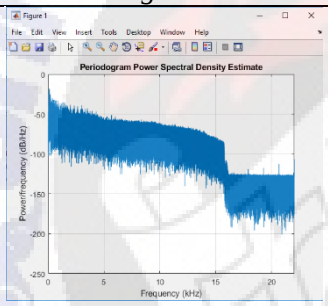
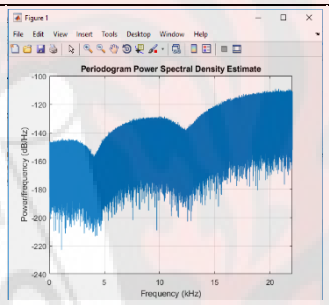
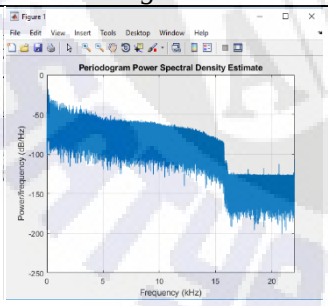
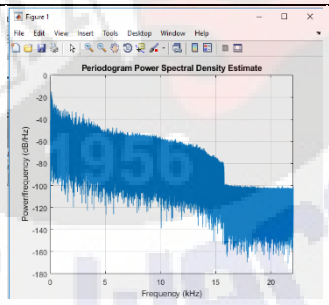
**Tabel 3** Pengujian Manipulasi *File* jpg

| No | Manipulasi Gambar *Robustness* | | *Recovery* | Hasil |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | | |
| 1. |  *Stego-data* |  *rotate* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |
| 2. |  *stego-data* |  *Cropping* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |
| 3. |  *Stego-data* |  *Noise* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |

Pengujian ketiga dilakukan *recovery* dengan diberikan serangan pada citra *stego-data*, pada bagian pertama terlihat perbandingan pada menu sebelum, *stego-data* dan pada menu sesudah *rotate* setelah diberikan serangan maka terlihat pada menu *recovery* data citra *stego-data* tidak bisa di *extract* dan data rahasia tidak bisa ditampilkan, pada bagian kedua *stego-data* diberikan serangan dengan *cropping* dan pada menu *recovery* data citra tidak berhasil di *extract* maka data *secret.txt* tidak dapat ditampilkan, selanjutnya pada bagian tiga *stego-data* diberikan serangan *noise* dan hasilnya masih terlihat sama pada bagian satu dan

bagian dua data citra tidak dapat di *extract*. Untuk pengujian terhadap *file* wav, ditunjukan pada Tabel 4.

**Tabel 4** Pengujian Manipulasi *File* wav

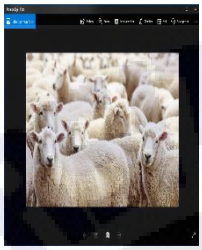
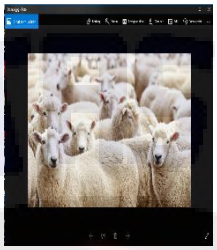
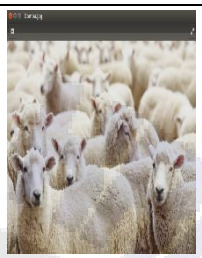
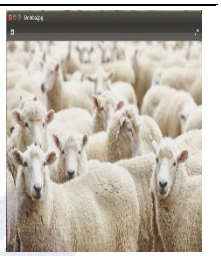
| No | Manipulasi Gambar *Robustness* | | *Recovery* | Hasil |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | | |
| 1. |  *Stego-data* |  *Reverse* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |
| 2. |  *Stego-data* |  *Silent* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |
| 3. |  *Stego-data* |  *Echo* | *extracting data…steghide : could not extract any data with that passphrase* | Tidak berhasil |

Pada Tabel 4, terlihat pengujian terhadap *file* wav. *Stego-data* pada menu sebelum dan menu sesudah terdapat histogram setelah diberi serangan, dengan bagian pertama serangan berupa *reverse* kemudian setelah itu *file* di *extract* dan perintah yang di hasilkan adalah *file* tidak berhasil di *extract*, bagian kedua diberikan serangan berupa *silent* terhadap *file* dan kemudian di *extract* dan hasilnya tidak bisa di *extract*, bagian ketiga tidak jauh beda dengan bagian yang pertama dan kedua, perbedaan hanya terlihat pada histogram, tetapi hasilnya tetap tidak dapat di *extract*.

Hasil analisa perbandingan dari kedua *platform* yaitu, *linux* dan *windows* akan dibahas bagaimana cara yang efektif dari dari perbandingan pada

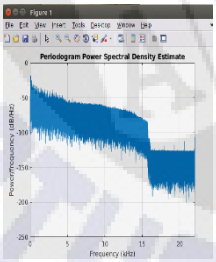
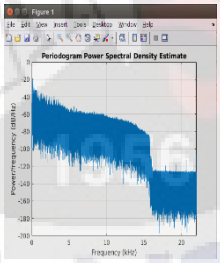
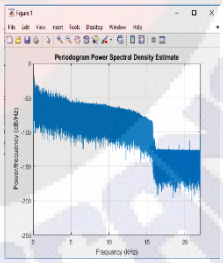
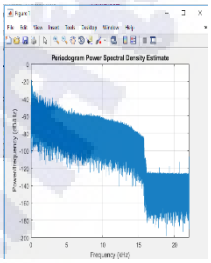
masing-masing *platform*. Perbandingan citra secara kasat mata ditunjukan pada Tabel 5.

**Tabel 5** Perbandingan Kasat Mata Domba.jpg

| ***File* Citra** | ***Windows*** | | ***Linux*** | |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | **Sebelum** | **Sesudah** |
| Domba.jpg |  |  |  |  |

Pada Tabel 5 tidak terdapat perbedaan secara kasat mata, tanpa adanya serangan dan di uji lebih jauh *file* tidak dapat diketahui. Maka secara kasat mata *file* dinyatakan aman. Perbandingan secara kasat mata juga dilakukan pada *file* wav, ditunjukan pada Tabel 6.

**Tabel 6** Perbandingan Kasat Mata Iwan.wav

| ***File* Citra** | ***Windows*** | | ***Linux*** | |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | **Sebelum** | **Sesudah** |
| Iwan.wav |  |  |  |  |

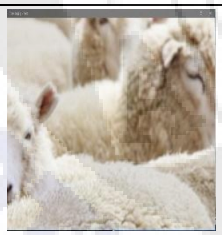
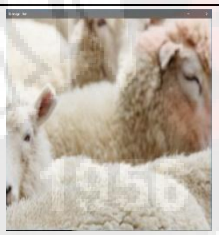
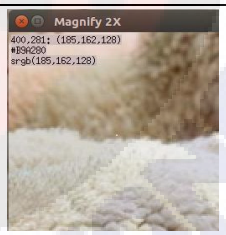
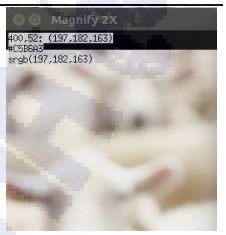
Pada Tabel 6, tampilan histogram yang dapat di uji dengan dilihat secara kasat mata, sebelumnya telah di uji dengan menggunkan pendengaran antara kedua *file* citra asli dan citra *stego* dan tidak ditemukan adanya perbedaan melalui pendengaran. Adanya perbedaan histogram antara kedua *file*. Maka *file* dinyatakan aman ketika didengar, jika ditampilkan histogram kedua citra maka akan ditemukan perbedaan. Perbandingan kapasitas citra ditunjukan pada Tabel 7.

**Tabel 7** Kapasitas Citra

| ***File* Citra** | **Sebelum** | | **Sesudah** | |
|---|---|---|---|---|
| | ***Windows*** | ***Linux*** | ***Windows*** | ***Linux*** |
| Domba.jpg | 117.054*bytes* | 117.054*bytes* | 136.479*bytes* | 119.482*bytes* |
| Iwan.wav | 52.185.767*bytes* | 52.185.767*bytes* | 52.185.767*bytes* | 52.185.767*bytes* |

Perbandingan kapasitas citra sebelum dan sesudah penyisipan pada *windows* sebelum penyisipan kedua citra Domba.jpg memiliki kapasitas yang sama, kemudian setelah disisipkan *file* rahasia terdapat perbedaan kapasitas, tetapi pada *linux* tidak terdapat banyak kapasitas yang diambil dibandingkan dengan di *windows*. Pada *file* citra Iwan.wav terlihat sebelum dan sesudah penyisipan pada masing-masing *platform* tidak terdapat perbedaan pada kapasitas citra. Maka penyisipan untuk *file* wav tergolong aman. Pengujian *fidelity*, *zooming* ditunjukan pada Tabel 8.

**Tabel 8** *Zooming* Kedua *Platform*

| | ***Windows*** | | ***Linux*** | |
|---|---|---|---|---|
| ***File* Citra** | **Sebelum** | **Sesudah** | **Sebelum** | **Sesudah** |
| Domba.jpg |  |  |   |   |

Pada Tabel 8, dilakukan pengujian *fidelity* dengan melakukan *zooming* pada kedua *platform*, terlihat bahwa *steghide* di *windows* tahan terhadap *zooming* dan di *linux* tidak tahan terhadap *zooming*, jika pihak lain melakukan *zooming* pada *file stego-data* di *linux* maka akan menimbulkan kecurigaan terhadap file *stego-data* berbeda jika dibandingkan dengan di *windows*.

Untuk perbandingan selanjutnya menggunkaan *robustness* dan *recovery* menggunkan *linux* dilakukan dengan *file* Domba.jpg dan Iwan.wav terlihat pada Tabel 1 dan Tabel 2. Menggunakan *windows* terlihat pada Tabel 3 dan Tabel 4. Setelah dilakukan pengujian pada kedua *platform file* tidak dapat di *extract.*

## 5. Simpulan dan Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut: (1) *Steghide* pada kedua *platform* dapat dilakukan penyisipan dan pengambilan data rahasia tanpa adanya banyak serangan; (2) *Steghide* pada *linux* tidak tahan terhadap serangan *fidelity*, *zooming*, namun pada *windows* proses *zooming* tidak ditemukan perbedaan; (3) Pada *linux* terdapat sedikit perbedaan kapasitas gambar yaitu, *cover-data* 117.054*bytes* dan *stego-data* 119.482*bytes*. Pada *windows* terdapat banyak perbedaan kapasitas *cover-data* 117.054*bytes* dan *stego-data* 136.479*bytes*; (4) Untuk *file* wav pada *windows* dan *linux* tidak ditemukan perbedaan suara pada citra *cover* dan citra *stego*, tetapi terdapat perbedaan histogram jika dilihat secara kasat mata; (5) Untuk *file* Domba.jpg tidak terdapat perbedaan secara kasat mata pada citra *cover* dan citra *stego*; (6) *File* tidak dapat di extract setelah dilakukan serangan *robustness* dan *recovery* pada *steghide*. Dari kesimpulan yang ada maka dapat ditambahkan saran sebagai berikut. (1) *Steghide* perlu adanya penambahan algoritma yang tahan terhadap serangan *robustness* dan *recovery*; (2) Perlu dilakukan penelitian yang lebih lanjut terhap *file* ekstensi lainnya yang belum di uji; (3) *Steghide* lebih efektif jika menggunkan *windows.*

## 6. Daftar Pustaka


[1] I Nyoman Piarsa, 2011, Steganografi Pada Citra JPEG Dengan Metode Sequential dan Spreading, *Lontar Komputer* 2: 1-2.

[2] Wahyu Hidayat, 2011, Mendeteksi Keberadaan Pesan Tersembunyi dalam Citra dengan Blind Steganalysis, *Conference Paper*: 1-3.

[3] Ratna Mutia S. 2017. *Studi dan Pengujian Algoritma Steganografi pada Aplikasi Steghide*. Bandung : Institut Teknologi Bandung.

[4] Ari Septayuda, Bambang Hidayat, Hilal Hudan Nuha, 2014, Analisis Steganografi Citra Digital menggunakan Metode Spread Spectrum Berbasis Android, *e-Proceeding of Engineering*, Vol.1:147-149.

[5] Stefanus Yerian Elandha, Magdalena A. Ineke Pakereng, M.Kom, 2016, Perancangan dan Implementasi Steganografi Menggunakan Metode Redundant Pattern Encoding dengan Algoritma AES (Advanced Encryption Standard),

[6] Muchlisin Riadi, 2017, Sejarah, Prinsip Kerja dan Teknik Steganografi, *https://www.kajianpustaka.com*. Diakses tanggal 20 Oktober 2018.

[7] R. Kristoforus JB, Stefanus Aditya BP, 2012, Implementasi Algoritma Rijndael untuk Enkripsi dan Dekripsi Pada Citra Digital, *Seminar Nasional Aplikasi Teknologi Informasi*.